

# CBSE Class 11 Revised Computer Science Syllabus 2020-21

## Computer Science

(Revised)  
CLASS-XI  
Code No. 083  
2020-21

### 1. Learning Outcomes

- Ability to understand and apply basic computational thinking.
- Ability to understand the notion of data types and data structures and apply in different situations.
- Ability to appreciate the notion of an algorithm and apply its structure including how algorithms handle corner cases.
- Ability to develop a basic understanding of computer systems - architecture, operating system, mobile and cloud computing.
- Ability to work in the cyber world with understanding of cyber ethics, cyber safety and cybercrime
- Ability to make use the value of technology in societies, gender and disability issues and the technology behind biometric ids.

### 2. Distribution of Marks

| Unit No. | Unit Name                                  | Theory Marks | Periods |           |
|----------|--|--------------|---------|-----------|
|          |  |              | Theory  | Practical |
| I        | Computer Systems and Organisation          | 10           | 10      | 5         |
| II       | Computational Thinking and Programming - 1 | 45           | 50      | 35        |
| III      | Society, Law and Ethics                    | 15           | 20      | ----      |
|          | Total                                      | 70           | 80      | 40        |

### Unit I: Computer Systems and Organisation

- Basic computer organisation: description of a computer system and mobile system, CPU, memory, hard disk, I/O, battery.
- Types of software: Application software, System software and Utility software.
- Memory Units: bit, byte, MB, GB, TB, and PB.
- Boolean logic: NOT, AND, OR, NAND, NOR, XOR, NOT, truth tables and De Morgan's laws, Logic circuits
- Number System: numbers in base 2, 8, 16 and binary addition.
- Encoding Schemes : ASCII, ISCII and Unicode

- Concept of Compiler and Interpreter
- Operating System (OS) - need for an operating system, brief introduction to functions of OS, user interface

## **Unit II: Computational Thinking and Programming - 1**

Introduction to Problem solving: Problem solving cycle - Analysing a problem, designing algorithms and representation of algorithm using flowchart and pseudo-code.

Familiarization with the basics of Python programming: a simple "hello world" program, the process of writing a program (Interactive & Script mode), running it and print statements; simple data-types: integer, float and string.

- Features of Python, Python Character Set, Token & Identifiers, Keywords, Literals, Delimiters, Operators.
- Comments: (Single line & Multiline/ Continuation statements), Clarity & Simplification of expression
- Introduce the notion of a variable and methods to manipulate it (concept of L-value and R-value even if not taught explicitly).
- Knowledge of data types and operators: accepting input from the console, assignment statement, expressions, operators and their precedence.
- Operators & types: Binary operators-Arithmetic, Relational Operators, Logical Operators, Augmented Assignment Operators.
- Execution of a program, errors- syntax error, run-time error and logical error.
- Conditional statements: if, if-else, if-elif-else; simple programs: e.g.: absolute value, sort 3 numbers and divisibility of a number.
- Notion of iterative computation and control flow: for(range(),len()), while, using flowcharts, suggested programs: calculation of simple and compound interests, finding the factorial of a positive number etc.
- Strings: Traversal, operations – concatenation, repetition, membership; functions/methods–len(), capitalize(), title(), upper(), lower(), count(), find(), index(), isalnum(), islower(), isupper(), isspace(), isalpha(), isdigit(), split(), partition(), strip(), lstrip(), rstrip(), replace(); String slicing.
- Lists: Definition, Creation of a list, Traversal of a list. Operations on a list - concatenation, repetition, membership; functions/methods–len(), list(), append(), extend(), insert(), count(), index(), remove(), pop(), reverse(), sort(), min(), max(), sum(); Lists Slicing; Nested lists; finding the maximum, minimum, mean of numeric values stored in a list; linear search on list of numbers and counting the frequency of elements in a list.

- Tuples: Definition, Creation of a Tuple, Traversal of a tuple. Operations on a tuple - concatenation, repetition, membership; functions/methods – len(), tuple(), count(), index(), sorted(), min(), max(), sum(); Nested tuple; Tuple slicing; finding the minimum, maximum, mean of values stored in a tuple; linear search on a tuple of numbers, counting the frequency of elements in a tuple.
- Dictionary: Definition, Creation, Accessing elements of a dictionary, add an item, modify an item in a dictionary; Traversal, functions/methods – len(), dict(), keys(), values(), items(), get(), update(), del(), del, clear(), fromkeys(), copy(), pop(), popitem(), setdefault(), max(), min(), count(), sorted() copy(); Suggested programs : count the number of times a character appears in a given string using a dictionary, create a dictionary with names of employees, their salary and access them.
- Introduction to Python modules: Importing math module (pi, e, sqrt, ceil, floor, pow, fabs, sin, cos, tan); random module (random, randint, randrange), statistics module (mean, median, mode).

### **Unit III: Society, Law and Ethics**

- Cyber safety: safely browsing the web, identity protection, confidentiality, social networks, cyber trolls and bullying.
- Appropriate usage of social networks: spread of rumours, and common social networking sites (Twitter, LinkedIn, and Facebook) and specific usage rules.
- Safely accessing web sites: adware, malware, viruses, trojans
- Safely communicating data: secure connections, eavesdropping, phishing and identity verification.
- Intellectual property rights, plagiarism, digital rights management, and licensing (Creative Commons, GPL and Apache), open source, open data, privacy.
- Privacy laws, fraud; cyber-crime- phishing, illegal downloads, child pornography, scams; cyber forensics, IT Act, 2000.
- Technology and society:
  - understanding of societal issues and cultural changes induced by technology.
  - E-waste management: proper disposal of used electronic gadgets.
  - Identity theft, unique ids and biometrics.
  - Gender and disability issues while teaching and using computers.

### 3. Practical

| S.No. | Area   | Marks<br>(Total=30) |
|-------|--|---------------------|
| 1.    | <b>Lab Test (12 marks)</b>   |                     |
|       | Python program (60% logic + 20% documentation + 20% code quality)  | <b>12</b>           |
| 2.    | <b>Report File + Viva (10 marks)</b>   |                     |
|       | Report file: Minimum 20 Python programs  | <b>7</b>            |
|       | Viva voce  | <b>3</b>            |
| 3.    | <b>Project (8 marks)</b><br>(that uses most of the concepts that have been learnt See CS-XII for the rules regarding the projects) |                     |

### 4. Suggested Practical List

#### Python Programming

- Input a welcome message and display it.
- Input two numbers and display the larger / smaller number.
- Input three numbers and display the largest / smallest number.
- Given two integers x and n, compute  $x^n$ .
- Write a program to input the value of x and n and print the sum of the following series:

- $1+x+x^2+x^3+x^4+ \dots x^n$
- $1-x+x^2-x^3+x^4+ \dots x^n$
- $x + \frac{x^2}{2} - \frac{x^3}{3} + \frac{x^4}{4} + \dots \frac{x^n}{n}$
- $x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} + \dots \frac{x^n}{n!}$

- Determine whether a number is a perfect number, an armstrong number or a palindrome.
- Input a number and check if the number is a prime or composite number.
- Display the terms of a Fibonacci series.
- Compute the greatest common divisor and least common multiple of two integers.
- Count and display the number of vowels, consonants, uppercase, lowercase characters in string.
- Input a string and determine whether it is a palindrome or not; convert the case of characters in a string.
- Find the largest/smallest number in a list/tuple
- Input a list of numbers and swap elements at the even location with the elements at the odd location.
- Input a list/tuple of elements, search for a given element in the list/tuple.
- Input a list of numbers and test if a number is equal to the sum of the cubes of its digits. Find the smallest and largest such number from the given list of numbers.
- Create a dictionary with the roll number, name and marks of n students in a class and display the names of students who have marks above 75.